



Atividade: Implementando o Padrão Composite



Objetivo da Aula

Implementar o **padrão Composite** para criar uma estrutura de pedidos com:

- **Itens individuais** (folhas)
- **Caixas** que podem conter outros itens ou caixas (compostos)



Estrutura do Código

1 Interface PedidoComponente

PedidoComponente.java

```
public interface PedidoComponente {  
    double getPreco();  
    void exibir(String prefixo);  
}
```

2 Classe Item (Complete!)

Item.java

```
public class Item implements PedidoComponente {  
    private String nome;  
    private double preco;  
  
    // Construtor (Complete!)  
    public Item(String nome, double preco) {
```

```

    }

    @Override
    public double getPreco() {
        return _____;
    }

    @Override
    public void exibir(String prefixo) {
        System.out.println(prefixo + "- " + nome + ": R$ " + preco);
    }
}

```

3 Classe caixa (Complete!)

Caixa.java

```

import java.util.ArrayList;
import java.util.List;

public class Caixa implements PedidoComponente {
    private String descricao;
    private List<PedidoComponente> conteudo = new ArrayList<>();

    // Construtor (Complete!)
    public Caixa(String descricao) {
        _____
    }

    public void adicionar(PedidoComponente componente) {
        conteudo.add(componente);
    }

    @Override
    public double getPreco() {
        double total = 0;
        for (PedidoComponente c : conteudo) {
            total += _____;
        }
        // Exibe o total desta caixa
        System.out.println(descricao + " - Total: R$" + total);
        return _____;
    }

    @Override

```

```

    public void exibir(String prefixo) {
        System.out.println(prefixo + "+ Caixa: " + descricao);
        for (PedidoComponente c : conteudo) {
            c.exibir("        ");
        }
    }
}

```



Exemplo de Uso

Main.java

```

public class Main {
    public static void main(String[] args) {
        Item item1 = new Item("Notebook", 3500.00);
        Item item2 = new Item("Mouse", 120.00);

        Caixa caixaPequena = new Caixa("Eletrônicos");
        caixaPequena.adicionar(item1);
        caixaPequena.adicionar(item2);

        Caixa caixaGrande = new Caixa("Pedido Completo");
        caixaGrande.adicionar(caixaPequena);
        caixaGrande.adicionar(new Item("Cupom Fiscal", 0.00));

        System.out.println("=== Hierarquia do Pedido ===");
        caixaGrande.exibir("");

        System.out.println("\n=== Totais ===");
        double totalFinal = caixaGrande.getPreco();
        System.out.println("TOTAL GERAL: R$" + totalFinal);
    }
}

```

Saída Esperada

```

=== Hierarquia do Pedido ===
+ Caixa: Pedido Completo
  + Caixa: Eletrônicos

```

```
- Notebook: R$3500.0
- Mouse: R$120.0
- Cupom Fiscal: R$0.0

=== Totais ===
Eletrônicos - Total: R$3620.0
Pedido Completo - Total: R$3620.0
TOTAL GERAL: R$3620.0
```



Exercício Proposto

1. Complete os métodos `getPreco()` e `exibir()` na classe `Item`.
2. Implemente a soma de preços e exibição hierárquica na classe `Caixa`.
3. Teste criando um pedido com:
 - Uma **caixa** com 2 itens
 - Uma **segunda caixa** contendo a primeira e mais um item



Dicas

- **Item:** Retorne apenas o preço do item em `getPreco()`.
- **Caixa:** Percorra a lista `conteudo` somando preços ou exibindo cada elemento.
- **Identação:** Use `prefixo + " "` para itens dentro de caixas.